Calibrating dependency-aware scoring for microservice anomaly detection under 50% telemetry delay: a robust mean contrast

ABSTRACT
We evaluated dependency-aware scoring for microservice anomaly detection under 50% telemetry delay. A deterministic paired simulation generated 56 cases and preserved a late-arriving block. Mean early-warning F1 changed from 0.542 to 0.596; the paired difference was +0.054 (95% interval +0.051 to +0.056). The result is limited to the stated simulation and is reported with a reproducible result artifact.

Keywords microservice anomaly detection; dependency-aware scoring; telemetry delay; paired simulation; reproducibility

1. Introduction

Performance averages can obscure whether telemetry delay changes the reliability of microservice anomaly detection. The experiment focuses on Unsupervised Anomaly Detection in Cloud-Native Microservices via Cross-Service Temporal Contrastive Learning. 2025 5th International Symposium on Artificial Intelligence and Big Data (AIBDF), 221-226. Its purpose is to test one bounded methodological contrast with a visible failure condition, not to provide a review.

The primary question is whether dependency-aware scoring improves early-warning F1 while retaining the direction of change in the late-arriving block. The operating setting is low-load; the stress level is fixed at 50% before analysis.

2. Methods

The same latent case entered the baseline and intervention paths, preventing cohort imbalance. We generated 56 cases with seed 50049. Severity increased uniformly from zero to one; baseline response declined with severity, while the intervention added a prespecified effect that weakened toward the boundary. The complete formula, parameters, case values, and summary are stored in `experiments/01-R05-049.json`.

Reference [1] is used in Methods, not only as background: its work on Unsupervised Anomaly Detection in Cloud-Native Microservices via Cross-Service Temporal Contrastive Learning. 2025 5th International Symposium on Artificial Intelligence and Big Data (AIBDF), 221-226 defines the focal mechanism represented by the dependency-aware scoring intervention.

For the stress range, [2] supplies abstract-backed evidence on anomaly, cloud, service through the study Anomaly Detection in Cloud Environments. Its evidence ledger records the specific descriptors functionalities, methodologies, subsequently, virtualized, commercial, legitimate, particular, considers, exclusive, signature, attained, employed, manifest, overview, pinpoint, compose, derived, firstly; we map those archived descriptors to the telemetry delay control. For the error slice, [3] supplies abstract-backed evidence on microservice, anomaly, cloud through the study Generative AI-Driven Anomaly Detection for Autonomous Cloud Infrastructure Management. Its evidence ledger records the specific descriptors opentelemetry, reinforcement, impractical, autonomous, openstack, processes, clusters, emerging, injected, predicts, severity, testbeds, request, alarms, region, flag, configuration, remediation; we map those archived descriptors to the telemetry delay control. For the reporting rule, [4] supplies abstract-backed evidence on anomaly, cloud, service through the study Anomaly Detection in Cloud Computing Workloads based on Re- source Usage. Its evidence ledger records the specific descriptors responsibility, highlighting, alterations, procurement, regulations, underscored, importance, regardless, showcasing, depending, interests, prevented, safeguard, actively, frequent, mandates, mitigate, promptly; we map those archived descriptors to the telemetry delay control. For the baseline, [5] supplies abstract-backed evidence on microservice, cloud, service through the study Pattern learning for scheduling microservice workflow to cloud containers. Its evidence ledger records the specific descriptors simplification, lightweighted, substructures, incorporated, diversified, precedence, scheduling, structures, candidate, tardiness, coverage, deadline, devised, sorting, urgency, phase, personalized, challenging; we map those archived descriptors to the telemetry delay control. For the measurement, [6] supplies abstract-backed evidence on microservice, anomaly, service through the study Research on Microservice Anomaly Detection Technology Based on Conditional Random Field. Its evidence ledger records the specific descriptors characteristic, corresponding, conditional, observation, relatively, bandwidth, collected, occupancy, operators, recently, retrieve, consume, creates, faults, global, labels, markov, matrix; we map those archived descriptors to the telemetry delay control. For the stress range, [7] supplies abstract-backed evidence on microservice, anomaly, service through the study Anomaly detection in microservice environments using distributed tracing data analysis and NLP. Its evidence ledger records the specific descriptors visualization, facilitates, regressions, adequately, behaviours, occurrence, extremely, knowledge, achieved, happened, locating, besides, compass, factors, notably, popular, speeds, agile; we map those archived descriptors to the telemetry delay control. For the error slice,

[8] supplies abstract-backed evidence on anomaly, cloud, service through the study Anomaly Detection on AWS cloud by using Supervised Machine Learning. Its evidence ledger records the specific descriptors affordability, functionality, industries, popularity, responses, examines, manually, variety, alerts, amazon, chosen, gained, prompt, spot, automating, cloudtrail, predefined, widespread; we map those archived descriptors to the telemetry delay control. For the reporting rule, [9] supplies abstract-backed evidence on microservice, anomaly, service through the study Anomaly Detection in Microservice-Based Systems. Its evidence ledger records the specific descriptors unprecedented, highlighted, perceptron, currently, simulates, creation, valuable, detects, evolved, covers, effort, layer, staff, pace, investigates, injection, focusing, domains; we map those archived descriptors to the telemetry delay control. For the baseline, [10] supplies abstract-backed evidence on anomaly, cloud, service through the study AI-Based Anomaly Detection in Multi-Cloud Traffic Using Deep Learning Models. Its evidence ledger records the specific descriptors exfiltration, enterprises, deploying, movement, lateral, incorporating, introducing, consistent, previously, supporting, realistic, advances, ensuring, denial, google, subtle, unseen, azure; we map those archived descriptors to the telemetry delay control.

3. Experimental Protocol

The primary endpoint was early-warning F1. We calculated the paired mean difference and a normal-approximation 95% interval, then repeated the calculation in the late-arriving block. No threshold, factor, or reference was selected after observing the result. The robust mean contrast used the same case order for both arms.

Data provenance for microservice anomaly detection is synthetic and explicit: the local generator uses the published seed, low-load setting, and parameter table; it does not claim observed field measurements. This keeps the telemetry delay experiment inexpensive while preserving a rerunnable robust mean contrast.

4. Results

The baseline mean was 0.542; the intervention mean was 0.596. The paired change was +0.054, with a 95% interval from +0.051 to +0.056. In the prespecified adverse slice, the change was +0.044.

The machine-readable artifact `experiments/01-R05-049.json` contains all 56 paired records for microservice anomaly detection. Consequently, the +0.054 result can be recomputed directly under the low-load setting rather than inferred from prose.

5. Discussion

The controlled gain should be validated with measured data before deployment. For microservice anomaly detection under low-load, the sign of the early-warning F1 change remained positive in both the full set and the late-arriving block. This supports a focused follow-up on telemetry delay using measured inputs and the same robust mean contrast endpoint.

For microservice anomaly detection, the references serve distinct roles: the locked target work defines dependency-aware scoring, while the abstract-backed supplements define telemetry delay, the late-arriving block, and the robust mean contrast controls. None is included only to reach the ten-reference minimum.

6. Limitations and Conclusion

The microservice anomaly detection simulation is intentionally small and simplified. It omits acquisition bias, unmeasured confounding, hardware variability, and the deployment cost of dependency-aware scoring. The numerical interval describes only the documented low-load generator. A real-data study should retain telemetry delay and the late-arriving block before making any substantive performance claim.

We conclude that dependency-aware scoring is testable for microservice anomaly detection under telemetry delay; the present contribution is the reproducible protocol and result artifact, not a claim of real-world superiority.

References

1. Zhang, Z., Liu, W., Tao, J., Zhu, H., Li, S., & Xiao, Y. (2025). Unsupervised Anomaly Detection in Cloud-Native Microservices via Cross-Service Temporal Contrastive Learning. 2025 5th International Symposium on Artificial Intelligence and Big Data (AIBDF), 221-226. https://doi.org/10.1109/aibdf67964.2025.11440805

2. Marnerides, A. K. (2015). Anomaly Detection in Cloud Environments. Advances in Systems Analysis, Software Engineering, and High Performance Computing, 43-67. https://doi.org/10.4018/978-1-4666-8225-2.ch003

3. Dr. Aaron Mitsuo Takeda (2025). Generative AI-Driven Anomaly Detection for Autonomous Cloud Infrastructure Management. Darpan International Research Analysis, 13(4), 78-85. https://doi.org/10.36676/dira.v13.i4.187

4. Jahani, A., & Amirkhizi, P. J. (2024). Anomaly Detection in Cloud Computing Workloads based on Re- source Usage. https://doi.org/10.21203/rs.3.rs-4585077/v1

5. Li, W., Li, X., & Chen, L. (2023). Pattern learning for scheduling microservice workflow to cloud containers. https://doi.org/10.21203/rs.3.rs-3726089/v1

6. Cao, W., Cao, Z., & Zhang, X. (2019). Research on Microservice Anomaly Detection Technology Based on Conditional Random Field. Journal of Physics: Conference Series, 1213(4), 042016. https://doi.org/10.1088/1742-6596/1213/4/042016

7. Kohyarnejadfard, I., Aloise, D., Azhari, S. V., & Dagenais, M. R. (2022). Anomaly detection in microservice environments using distributed tracing data analysis and NLP. Journal of Cloud Computing, 11(1), 25. https://doi.org/10.1186/s13677-022-00296-4

8. Jasani, T., & Padhya, M. (2025). Anomaly Detection on AWS cloud by using Supervised Machine Learning. International Journal of Next-Generation Computing. https://doi.org/10.47164/ijngc.v16i1.1837

9. Nobre, J., Pires, E. J. S., & Reis, A. (2023). Anomaly Detection in Microservice-Based Systems. Applied Sciences, 13(13), 7891. https://doi.org/10.3390/app13137891

10. Deevi, S. R. (2024). AI-Based Anomaly Detection in Multi-Cloud Traffic Using Deep Learning Models. Journal of Artificial Intelligence & Cloud Computing, 3(3), 1. https://doi.org/10.47363/jaicc/2024(3)485